Homework 1: SF2852: Optimal Control

Grading: You may use $\min\left(2, \frac{\text{your credit}}{10}\right)$ extra points on the exam.

(A) Note that the total credit in this homework set is more than you need to get full bonus.

(B) You need to turn in your own solution.

Problem 1

Solve the following optimal control problems using dynamic programming

$$\begin{aligned} \min \quad & 2(x_2+1)^2 + (u_0^2 + u_1^2) \\ \text{subject to} \quad & x_{k+1} = x_k + u_k; \qquad \text{for } k = 0, 1, \text{ and } x_0 = 1. \end{aligned}$$

Determine the minimal cost and the corresponding u_0, u_1, x_1, x_2..... (4p)

Problem 2

A decision maker must choose between two activities over a time interval $[0, t_f]$. Each activity earns a reward at rate $g_k(t)$, $k = 1, 2$. Every switch between the two activities costs $c > 0$. As an example, the reward for starting with activity 1, switch to activity 2 at time t_1 and back to 1 at time $t_2 > t_1$ earns the total reward

$$\int_0^{t_1} g_1(t)dt + \int_{t_1}^{t_2} g_2(t)dt + \int_{t_2}^{t_f} g_1(t)dt - 2c$$

We want to find a switching sequence that maximize the total reward. Switching can only occur inside the the interval $(0, t_f)$.

Assume the function $g_1(t) - g_2(t)$ changes sign a finite number of times in the interval $(0, t_f)$.

(a.) Formulate the problem as a sequential optimization problem and then formulate the corresponding DP recursion..... (3p)

(b.) Solve the dynamic programming problem in (a) using the DP recursion for the case when $c = 2$ and

$$g_1(t) = \begin{cases} 4, & 0 \le t < 1 \\ 0, & 1 \le t < 2 \\ 5, & 2 \le t \le 3 \end{cases}, \qquad g_2(t) = \begin{cases} 1, & 0 \le t < 1 \\ 6, & 1 \le t < 2 \\ 2, & 2 \le t \le 3 \end{cases}$$

..... (3p)

Problem 3

Consider the system in Figure 1.

(a) Determine a diagonal state space realization of the linear system (the A matrix should be diagonal)

$$\dot{x} = Ax + Bu,\ x(0) = 0$$
$$y = Cx$$

Hint: Do a partial fraction expension of $G(s) = \dfrac{s-1}{s^2+3s+2}$

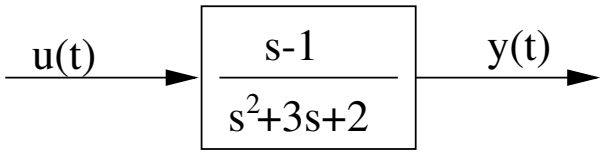

Figure 1: A signal is sent through a filter with transfer function $G(s) = \frac{s-1}{s^2+3s+2}$.

..... (2p)

(b) Assume the control is constrained as $|u(t)| \leq 1$. Determine an explicit expression for the optimal input (as a function of t_f) such that the output $y(t_f)$ is maximized, i.e. solve

$$\max Cx(t_f) \quad \text{subject to} \quad \begin{cases} \dot{x} = Ax + Bu,\ x(0) = 0 \\ |u(t)| \leq 1 \end{cases}$$

..... (4p)

Problem 4

The purpose of this problem is to balance an inverted pendulum using Model Predictive Control (MPC). The pendulum is subject to the gravity force and should be stabilized in the upward vertical position controlling the steering torque without using too much power.

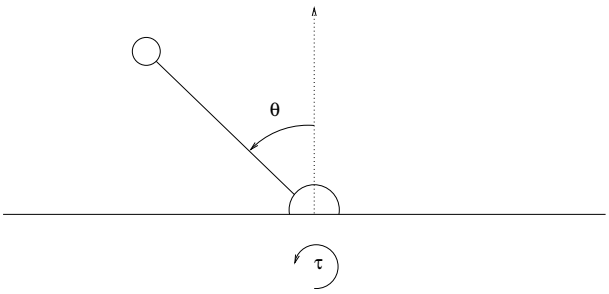

Figure 2: The inverted pendulum. Here τ is the torque and θ is the angular position.

In this exercise an approximated discrete-time model will be considered namely

$$\begin{aligned} \mathbf{z}_{k+1} &= \mathbf{\Phi}\mathbf{z}_k + \mathbf{\Gamma}u_k \\ y_k &= \mathbf{C}\mathbf{z}_k \end{aligned} \tag{1}$$

where T is the sampling time and the control signal is such that $\tau(t) = u_k$ for $t \in [kT, (k+1)T]$. Furthermore we have

$$\begin{aligned} \mathbf{z}_k &= \begin{bmatrix} \theta(kT) \\ \dot{\theta}(kT) \end{bmatrix}, \quad \mathbf{\Phi} = \begin{bmatrix} 1 & T \\ \alpha T & 1 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \\ \mathbf{C} &= \begin{bmatrix} 1 & 0 \end{bmatrix} \end{aligned}$$

where α is a constant that depends on the physical parameters. This model can be seen as the first order approximation of the small angle dynamic for discrete time. Hereafter it will be assumed to be the true model.

Considering the control purpose, a natural optimization criterion is the following linear quadratic problem

$$\begin{aligned} \min \quad & q|\mathbf{z}_{t+N|t}|^2 + \textstyle\sum_{k=0}^{N-1}(q|\mathbf{C}\mathbf{z}_{t+k|t}|^2 + r u_{t+k|t}^2) \\ \text{s.t.} \quad & \mathbf{z}_{t+k+1|t} = \mathbf{\Phi}\mathbf{z}_{t+k|t} + \mathbf{\Gamma}u_{t+k|t} \quad k = 0, 1, \ldots, N-1 \end{aligned} \tag{2}$$

where q and r are positive weights.

In the following problems we ask you to implement the above quadratic optimization problem for MPC of the inverted pendulum process. You should also experiment with the algorithm.

For a) and b) there are two options for the implementation:

1. Implement this using Matlab's function Quadprog. The appendix contains a Matlab skeleton that defines a suggested structure for your code. There are also some hints on how to implement the system matrices.

2. You can also write this using CVX as in HW#0.

(a) Solve the problem (2) using Matlab using the following parameter $\alpha = 0.5$, initial condition $\mathbf{z}_{0|0} = \begin{pmatrix} 0.5 & 1 \end{pmatrix}^T$, $r = 1$, the sampling time $T = 0.1$ and

(i) $N = 5$, $q = 5$,
(ii) $N = 10$, $q = 5$,
(iii) $N = 10$, $q = 1$.

What conclusions can you make regarding the convergence to the origin. ..... (4p)

Hint for the Matlab quadprog implementation: Rewrite the optimization problem (2) on the form

$$\begin{aligned} \min \quad & \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} \\ s.t. \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \end{aligned}$$

where $\mathbf{x} = \begin{pmatrix} \mathbf{z}_{t+1|t}^T & \dots & \mathbf{z}_{t+N|t}^T & u_{t|t} & \dots u_{t+N-1|t} \end{pmatrix}^T$ *(note that* $\mathbf{z}_{t|t}$ *is given and is not a variable in the optimization problem). Note also that the right hand side* $\mathbf{b}$ *depends on the last measured state* $\mathbf{z}_{t|t}$ *and must be updated in every iteration of the MPC algorithm (see code in the appendix).*

(b) The actuator used to control the pendulum has limitations on the maximum torque applicable. Hence it is useful to take these constraints in the control problem. It can be formulated as

$$\begin{aligned} \min \quad & q_f|\mathbf{z}_{t+N|t}|^2 + \textstyle\sum_{k=0}^{N-1}(q|\mathbf{C}\mathbf{z}_{t+k|t}|^2 + ru_{t+k|t}^2) \\ \text{s.t} \quad & \begin{cases} \mathbf{z}_{t+k+1|t} = \mathbf{\Phi}\mathbf{z}_{t+k|t} + \mathbf{\Gamma}u_{t+k|t} \quad k = 0, \dots, N-1 \\ -1 \le u_{t+k|t} \le 1, \quad k = 0, \dots, N-1 \end{cases} \end{aligned} \tag{3}$$

and apply MPC with the following parameter values $\mathbf{z}_{0|0} = \begin{pmatrix} 0.5 & 1 \end{pmatrix}^T$, $r = 1$,

(i) $N = 5$, $q = 5$

(ii) $N = 10$, $q = 5$,

(iii) $N = 10$, $q = 1$.

What are your conclusions?. ..... (2p)

Hint for the Matlab quadprog implementation: rewrite the problem on the form

$$\begin{aligned} \min \quad & \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}_{eq}\mathbf{x} = \mathbf{b}_{eq} \\ & \mathbf{A}\mathbf{x} \leq \mathbf{b}. \end{aligned}$$

(c) The predicted states can be written as a function of $u_{t+k|t}, \ldots, u_{t|t}$. Hence, one may define an optimization problem in the reduced variable vector $\mathbf{x} = \begin{pmatrix} u_{t|t} & \ldots & u_{t+N-1|t} \end{pmatrix}^T$. Formulate the optimization problem on the form

$$\begin{aligned} \min \quad & \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} + \mathbf{x}^T\mathbf{f} + \mathbf{g} \\ \text{s.t.} \quad & \mathbf{A}_{eq}\mathbf{x} = \mathbf{b}_{eq} \\ & \mathbf{A}\mathbf{x} \leq \mathbf{b} \end{aligned}$$

You don't need to implement this in Matlab. If you selected to use the CVX formulation to solve a) and b), then the hints in those problems may be useful to get started on this problem. ..... (2p)

Problem 5

Implement a solution of the Knapsack problem,

$$\max \sum_{j=0}^{n-1} p_j u_j$$
$$\text{subject to} \sum_{j=0}^{n-1} w_j u_j \leq W, \quad u_j = 0 \text{ or } 1, \quad j = 0, \ldots, n-1,$$

using dynamic programming in, e.g., Matlab. Assume that $W \in \mathbf{N}$ and $w_j \in \mathbf{N}, p_j \in \mathbf{R}$ are given for $j = 0, 1, \ldots, n-1$. .....(4p)

Hint: Reformulate this as we did in class by introducing a state describing, e.g., the used space $x_k := \sum_{j=0}^{k-1} w_j u_j$. *The assumptions on the weights to be integers allows for representing all possible states as* $\{0, 1, \ldots, W\}$.

Good luck!

1 Matlab code, inverted pendulum

```
%
%------- Basic system model
%
clear;
T=0.1;
Phi=[1 T;0.5*T 1];
Gam=[T^2/2;T];
C=[1 0];
n=size(Phi,1);
m=size(Gam,2);
%
%------- Parameters --------
%
q=5;
r=1;
N=5;
z0=[0.5;1];

%
%------- Define matrices for the QP --------
%

For you to do!

%
%----- For problem 2 with inequalities ---------
%
A=[];
b=[];

%
%----- Cost ------------
%

For you to do!

%
%------- MPC algorithm ------
```

```
%

M=100;   %time horizon

zt=z0;
yvec=[];
uvec=[];
options=optimset('largescale','off');
for flcnt1=1:M
    beq=AA*zt; % The matrix AA defines how the last measured state
               % determines the right hand side in the equality constraint.
    x=quadprog(H,f,A,b,Aeq,beq,[],[],[],options);
    ut=x(n*N+1);
    zt=Phi*zt+Gam*ut;
    yvec=[yvec;C*zt];
    uvec=[uvec;ut];
end
tvec=T*(1:1:M);
subplot(3,1,1)  %For the other two sets of parameters you should change
                %the third index to 2 and 3, respectively.
plot(tvec,yvec,'-',tvec,uvec,'--')
grid
```

2 Hints for the implementation

Here follows some hints that simplifies the Matlab implementation of block matrices

1. Block diagonal matrices can be created using the command `blkdiag(A,B)`.
2. The command `kron(eye(N),M)` generates the block matrix

$$\begin{bmatrix} M & 0 & \dots & 0 \\ 0 & M & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & M \end{bmatrix}$$

Generally, we have

$$\text{kron}(A, M) = \begin{bmatrix} a_{11}B & a_{12}M & \dots & a_{1n}M \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}B & a_{n2}M & \dots & a_{nn}M \end{bmatrix}$$

3 Knapsack problem, example implementation

```
% Number of items
N=10;

% Total Weight allowed
W=50;

% Create random items
w=round(rand(N,1)*W/N^(1/2)+0.5);
p=W/N^(1/2)*rand(N,1);

Wgrid=0:W;

% Compute optimal cost to go and optimal control
J=zeros(W+1, N+1);
U_opt=zeros(W+1, N+1);

for k=N:-1:1

    % For you to do!!
    % Determine J in the kth stage
    % No choise if item is larger than remaining space
    % Two options if item is smaller than remaining space
    % Determine U_opt, i.e., the optimal control at each stage and position

end

% Compute optimal path from optimal control
U_path=zeros(size(U_opt));
opt_path=zeros(1,N);
opt_path(1)=1;
U_path(1,1)=1;
```

```
for k=1:N
    opt_path(k+1)=opt_path(k)+U_opt(opt_path(k),k)*w(k);
    U_path(opt_path(k+1),k+1)=1;
end

% Plot results
figure(1)
subplot(1,2,1)
imagesc(0:N,Wgrid, min(U_path+0.1*U_opt,1))
colorbar
xlabel('Item/Stage')
ylabel('Weight utilized')
title('Optimal control')

subplot(1,2,2)
imagesc(0:N,Wgrid, J)
colorbar
xlabel('Item/Stage')
ylabel('Weight utilized')
title('Cost to go')

figure(2)
%subplot(2,2,3)
plot(0:N-1, w, 'sk', 0:N-1, p, '*r')
legend('Weight', 'Value')
grid on
ylim([0,ceil(max([w(:);p(:)]))])
xlabel('Item')
ylabel('Value/Weight')
```